



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/824,092	04/14/2004	Ajay Kumar	5681-72300	6152
58467 7590 06/25/2009				
MHKKG/SUN				
P.O. BOX 398				
AUSTIN, TX 78767				
EXAMINER				
NGUYEN, CINDY				
ART UNIT		PAPER NUMBER		
2161				
NOTIFICATION DATE		DELIVERY MODE		
06/25/2009		ELECTRONIC		

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

patent\_docketing@intprop.com  
ptomhkk@gmail.com

### Office Action Summary

**Application No.**

10/824,092

**Applicant(s)**

KUMAR, AJAY

**Examiner**

CINDY NGUYEN

**Art Unit**

2161

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --  
**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 31 March 2009.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1-39 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-39 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO/SI/08)  
Paper No(s)/Mail Date \_\_\_\_\_
- 4) ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date \_\_\_\_\_
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: \_\_\_\_\_

## DETAILED ACTION

This is in response amendment filed 03/31/09.

### *Response to Arguments*

Applicant's arguments have been considered but are moot in view of the new ground(s) of rejection.

### *Claim Rejections - 35 USC § 102*

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

1) Claims 1-39 are rejected under 35 U.S.C. 102(e) as being anticipated by Sameer et al. ("Core Java Data Objects: chapter 3 "Getting Started with JDO" ), page 45-88, July 22, 2003, (hereafter Sameer), Note the publication date of the complete book is Sep 11, 2003.  
[http://www.informit.com/content/images/0131407317/samplechapter/0131407317\\_ch03.pdf](http://www.informit.com/content/images/0131407317/samplechapter/0131407317_ch03.pdf)

The applied reference has a common assignee with the instant application. Based upon the earlier effective U.S. filing date of the reference, it constitutes prior art under 35 U.S.C. 102(e). This rejection under 35 U.S.C. 102(e) might be overcome either by a showing under 37 CFR 1.132 that any invention disclosed but not claimed in the reference was derived from the inventor of this application and is thus not the invention "by another," or by an appropriate showing under 37 CFR 1.131.

Regarding claims 1, 14 and 27, Sameer discloses: a system, a method and a computer-accessible storage medium, comprising: a processor; and to implement a class structure based data object enhancer configured to:  
input one or more classes (see page 49, line 7);

analyze the structure of the one or more classes to determine a persistence structure specifying data fields of the one or more classes to be persisted (see page 53, lines 14-22);

generate one or more enhanced classes corresponding to the one or more classes such that an object of the one or more classes is enhanced to persist data of the data fields to be persisted according to the persistence structure object, wherein the generation of each of said one or more enhanced classes comprises adding to the corresponding one of said one or more classes, one or more calls to persist data fields as specified by the persistence structure (fig. 3-2, the byte code enhancement process to either update the existing class files or creates new ones... see page 49, fig. 3-2 and lines 6-12).

Regarding claims 2, 15 and 28, all the limitations of this claim have been noted in the rejection of claims 1, 14 and 27 above. In addition, Sameer disclose: wherein to analyze the structure of the one or more classes, the class structure based enhancer is configured to make one or more Java reflection calls to the one or more classes (Sameer using JDO implementation to retrieve (call) the fields of persistent objects and tracks modifications to the fields and writes these changes back to the datastore, JDO implementation is responsible for mapping the fields of the persistent objects to and from memory, it is inherent of using Java reflection for mapping the fields of the persistent objects, see page 46, section 3.1 How Does JDO work?, lines 16-20; page 53, section 3.3.2 "mapping a class to the datastore", lines 6-8).

Regarding claims 3, 16, 29, all the limitations of this claim have been noted in the rejection of claims 1, 14 and 27 above. In addition, Sameer disclose: wherein to analyze the structure of the one or more classes, the class structure based enhancer is configured to parse byte code of the one or more classes to determine class and field attributes (Sameer teaches how to define a Java class that can be used with a JDO implementation and using Byte code enhancement such as fig. 3.2).

Regarding claims 4, 17 and 30, all the limitations of this claim have been noted in the rejection of claims 1, 14 and 27 above. In addition, Sameer discloses: wherein the class structure based enhancer is further configured to generate metadata that includes the results of the analysis of the structure of the one or more classes (see page 52, section 3.3 Defining a Class, using JDO to create/generate a Author class persistence-capable, contains the JDO metadata for the class).

Regarding claims 5, 18, 31, all the limitations of this claim have been noted in the rejection of claims 4, 17 and 30 above. In addition, Sameer discloses: wherein the generated metadata is output explicitly as a metadata file (see page 52, section 3.3.1 JDO metadata, lines 1-2).

Regarding claims 6, 19 and 32, all the limitations of this claim have been noted in the rejection of claims 5, 18, 31 above. In addition, Sameer discloses: wherein the metadata file is an extensible markup language (XML) file (see page 52, section 3.3.1 JDO metadata, lines 1-2).

Regarding claims 7, 20 and 33, all the limitations of this claim have been noted in the rejection of claims 1, 14 and 27 above. In addition, Sameer discloses: wherein the persistence structure corresponds to the structure of the one or more classes (see

page 51, section 3.3 Defining a class, using JDO to create a persistence-capable Java class).

Regarding claims 8, 21, 34, all the limitations of this claim have been noted in the rejection of claims 1, 14 and 27 above. In addition, Sameer discloses wherein the persistence structure maps the data to be persisted to a single table in a database (see page 54, lines 3-7).

Regarding claims 9, 22 and 35, all the limitations of this claim have been noted in the rejection of claims 1, 14 and 27 above. In addition, Sameer discloses: wherein to determine a persistence structure for the data of the one or more classes the class structure based enhancer is configured to apply one or more rules to the results of Java reflection calls of the one or more input class (Sameer using JDO implementation to retrieve (call) the fields of persistent objects and tracks modifications to the fields and writes these changes back to the datastore, JDO implementation is responsible for mapping the fields of the persistent objects to and from memory, it is inherent of using Java reflection for mapping the fields of the persistent objects, see page 46, section 3.1 How Does JDO work?, lines 16-20; page 53, section 3.3.2 "mapping a class to the datastore", lines 6-8).

Regarding claims 10, 23 and 36 all the limitations of this claim have been noted in the rejection of claims 9, 22 and 35 above. In addition, Sameer discloses: wherein the one or more rules applied by the class structure based enhancer include persisting class fields that are not static or transient (see page 77, section 3.9.6 Modifiers, lines 3-9).

Regarding claims 11, 24, 37, all the limitations of this claim have been noted in the rejection of claims 1, 14 and 27 above. In addition, Sameer discloses wherein the rules applied by the class structure based enhancer include storing persistent fields of a given class in a table corresponding to that class in a database (see page 54, lines 3-7).

Regarding claims 12, 25 and 38, all the limitations of this claim have been noted in the rejection of claims 1, 14 and 27 above. In addition, Sameer disclose: wherein the one or more classes are comprised in a Java Archive (JAR) file (Java metadata file see page 52, section 3.3.1 JDO Metadata, lines 5).

Regarding claims 13, 26, 39, all the limitations of this claim have been noted in the rejection of claims 1, 14 and 27 above. In addition, Sameer discloses, wherein the class structure based enhancer is further configured to output the enhanced one or more classes and a database schema for storing the data to be persisted in a persistent data store (see page 53, section 3.3.2 Mapping a class to the datastore, lines 5-8).

2) Claims 1-39 are rejected under 35 U.S.C. 102(b) as being anticipated by Lipkin et al. (US 20020120859, hereafter Lipkin).

Regarding claims 1, 14 and 27, Lipkin discloses: a system, a method and a computer accessible storage medium, comprising: a processor, (i.e., CPU 207, fig. 2, Lipkin); and

Memory (209, fig. 2) coupled to the processor and configured to store program instructions executable by the processor (paragraphs 0018, 0220, Lipkin) to implement a class structure based data object enhancer configured to (i.e., base class called Saba Object, paragraph 0254, lines 3, Lipkin):

input one or more classes (i.e., classes is registered in the meta-data store, paragraph 0255 (lines 2-3); 0256, Lipkin);

analyze the structure of the one or more classes to determine a persistence structure specifying data fields of the one or more classes to be persisted (i.e., using persistence algorithms to determine every Saba objects (class) and subclass and its metadata...determine whether the object invokes an insert or update method... see paragraphs 0253-0281 ... see paragraphs 0282-0295);

generate one or more enhanced classes corresponding to the one or more classes such that an object of the one or more classes is enhanced to persist data of the data fields to be persisted according to the persistence structure object (i.e., An administrative user interface is provided by which the meta-data definition of a given **class can be extended by adding (or removing) custom attributes** as needed (as enhanced classes). For each custom attribute, the user only needs to provide some very basic information **about the type of the field**, whether or not it is required, constraining minimum and maximum values for numeric fields, and a constraining list if the field is to be validated against a list of possible values ... see paragraphs 0284, lines 1-9; 0295-0295), wherein the generation of each of said one or more enhanced classes comprises adding to the corresponding one of said one or more classes, one or more calls to persist data fields as specified by the persistence structure (i.e., The persistence framework defines a common code path used to **create new objects**, restore and update existing objects, delete objects, and find objects. The code path consists of a set of Java code and **database stored procedures to construct and verify object data** and SQL commands to save and restore information using a relational database,... see paragraphs 0231, 0235-0237, 0248, 0284, 0286, Lipkin).

Regarding claims 2, 15 and 28, all the limitations of this claim have been noted in the rejection of claims 1, 14 and 27 above. In addition, Lipkin disclose: wherein to analyze the structure of the one or more classes, the class structure based enhancer is configured to make one or more Java reflection calls to the one or more classes (i.e., some of the metadata that is currently captured about class or an attributed could be

Art Unit: 2161

dynamically determined using the java rejection API...see paragraph 0350, lines 1-4, Lipkin).

Regarding claims 3, 16, 29, all the limitations of this claim have been noted in the rejection of claims 1, 14 and 27 above. In addition, Lipkin discloses: wherein to analyze the structure of the one or more classes, the class structure based enhancer is configured to parse byte code of the one or more classes to determine class and field attributes ( i.e., parse the query using RQL parser...see paragraphs 1111, 1125, Lipkin).

Regarding claims 4, 17 and 30, all the limitations of this claim have been noted in the rejection of claims 1, 14 and 27 above. In addition, Lipkin discloses: wherein the class structure based enhancer is further configured to generate metadata that includes the results of the analysis of the structure of the one or more classes (i.e., An administrative user interface is provided by which the meta-data definition of a given class can be extended by adding (or removing) custom attributes as needed, see paragraph 0284, Lipkin).

Regarding claims 5, 18, 31, all the limitations of this claim have been noted in the rejection of claims 4, 17 and 30 above. In addition, Lipkin discloses: wherein the generated metadata is output explicitly as a metadata file (i.e., The jar file can be created using the jar application supplied with JDK, or by using some GUI front-end utility provided by the J2EE server being used,... see paragraph 0426, Lipkin).

Regarding claims 6, 19 and 32, all the limitations of this claim have been noted in the rejection of claims 5, 18, 31 above. In addition, Lipkin discloses: wherein the metadata file is an extensible markup language (XML) file (i.e., XML file,... see paragraph 0419, Lipkin).

Regarding claims 7, 20 and 33, all the limitations of this claim have been noted in the rejection of claims 1, 14 and 27 above. In addition, Lipkin discloses: wherein the persistence structure corresponds to the structure of the one or more classes (i.e., The persistence framework is highly flexible because it is metadata-driven. For each class of object, the system provides a set of metadata--data about data--that defines the class' properties and behavior,... see paragraph 0231, Lipkin).

Regarding claims 8, 21, 34, all the limitations of this claim have been noted in the rejection of claims 1, 14 and 27 above. In addition, Lipkin discloses wherein the persistence structure maps the data to be persisted to a single table in a database (i.e., two objects to a single table... see paragraphs 0278, 0293-0295, Lipkin).

Regarding claims 9, 22 and 35, all the limitations of this claim have been noted in the rejection of claims 1, 14 and 27 above. In addition, Lipkin discloses: wherein to determine a persistence structure for the data of the one or more classes the class structure based enhancer is configured to apply one or more rules to the results of Java

reflection calls of the one or more input class (i.e., The BDK provides a mechanism to enable and disable these business rules. For example, a customer can configure whether a manager's approval is required to register for a class. Similar business rules can be handled for other types of applications, see paragraphs 0305, Lipkin)..

Regarding claims 10, 23 and 36 all the limitations of this claim have been noted in the rejection of claims 9, 22 and 35 above. In addition, Lipkin discloses: wherein the one or more rules applied by the class structure based enhancer include persisting class fields that are not static or transient (i.e., the timestamp is updated whenever the state of an object is altered and the object is successfully committed to persistent storage, see paragraph 0301, Lipkin).

Regarding claims 11, 24, 37, all the limitations of this claim have been noted in the rejection of claims 1, 14 and 27 above. In addition, Lipkin discloses wherein the rules applied by the class structure based enhancer include storing persistent fields of a given class in a table corresponding to that class in a database (see paragraph 0295, Lipkin).

Regarding claims 12, 25 and 38, all the limitations of this claim have been noted in the rejection of claims 1, 14 and 27 above. In addition, Lipkin discloses: wherein the one or more classes are comprised in a Java Archive (JAR) file (Java files...see paragraph 0426, Lipkin).

Regarding claims 13, 26, 39, all the limitations of this claim have been noted in the rejection of claims 1, 14 and 27 above. In addition, Lipkin discloses, wherein the class structure based enhancer is further configured to output the enhanced one or more classes and a database schema for storing the data to be persisted in a persistent data store (i.e., Automatic persistence service provided by the application server enhances the productivity of bean developers, is more efficient at runtime, and allows the bean's definition to be independent of the type of data store used for persistence (e.g., a relational object-oriented or relational database). A component developer will be responsible for declaring part or all of the attributes of an entity bean as persistent in its deployment descriptor, and then mapping them to fields in a database at deployment time. The interface and mechanism of such mapping would depend upon the application server being used, see paragraph 0384, Lipkin).



Any inquiry concerning this communication or earlier communications from the examiner should be directed to Cindy Nguyen whose telephone number is 571-272-4025. The examiner can normally be reached on 8:30-5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Apu M. Mofiz can be reached on 571-272-4080. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/C. N./  
Examiner, Art Unit 2161

/A.O./

Primary Examiner, Art Unit 2160